



# Transfer Learning, Numpy+OpenCV

Grzegorz Madejski

Część I: Transfer Learning

Część II: Numpy + OpenCV

# Transfer Learning

- Przykład: Umiesz jeździć na rowerze, więc szybciej nauczysz się jeździć na motorze. Wcześniejszą wiedzę, wykorzystasz w zadaniu, które jest podobne.
- Transfer learning to technika w uczeniu maszynowym, polegająca na wykorzystywaniu **wcześniejsz zdobytej wiedzy** w rozwiązaniu **nowego problemu**.



# Transfer Learning

- Bierzemy model wytrenowany na dużym zbiorze(**pre-trained model**). Dotrenowujemy (**fine-tuning**) go na naszym zbiorze danych, który nie musi być duży.
- Zalety: oszczędzamy czas na trening, lepsza dokładność, brak konieczności posiadania dużej bazy danych.
- W sieci istnieje wiele wytrenowanych modeli (na github, huggingface, kaggle). Modele sieci neuronowych są też wbudowane w keras:  
<https://keras.io/api/applications/>



# Transfer Learning

- Skorzystajmy z ResNet, żeby zobaczyć czy poradzi sobie z klasyfikacją słonia i żyrafy.
- Klasy: <https://github.com/BigWZhu/ResNet50/blob/master/imagenet-classes.txt>

```
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.applications.resnet50 import preprocess_input,
decode_predictions
from tensorflow.keras.utils import load_img, img_to_array

import numpy as np

model = ResNet50(weights='imagenet')

img_path = 'elephant.jpg'
img = load_img(img_path, target_size=(224, 224))
x = img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
print('Predicted:', decode_predictions(preds, top=3)[0])

img_path = 'giraffe.jpg'
img = load_img(img_path, target_size=(224, 224))
x = img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
print('Predicted:', decode_predictions(preds, top=3)[0])
```



Część I: Transfer Learning

Część II: Numpy + OpenCV

# Numpy + OpenCV

- **Numpy:** bardzo wydajna biblioteka do obliczeń numerycznych (wektory, macierze)
- **OpenCV:** bardzo dobra biblioteka do przetwarzania obrazów
- Przejrzymy możliwości bibliotek na praktycznych przykładach ([demo0X.py](#) oraz [opencv-demo0X.py](#))

